# پروژه قناری فاز اول

## بردیا اردکانیان

## 9831072

## REFERENCES

### CORE TABLES

1. USER
2. TWEET
3. MESSAGE
4. HASHTAG

### MAPPING TABLES (MAP TWO OR MORE ENTRIES TOGETHER)

1. LOGIN
2. FOLLOWER
3. BLOCKED
4. LIKED
5. REPLY

### FUNCTIONS

1. Register (create new user)
2. Login
3. Login History
4. Post (tweet a new tweet)
5. Get Personal Tweets
6. Follow
7. Unfollow
8. Block
9. Unblock
10. Timeline (following activity)
11. User Activity
12. Add Reply
13. Show Replies
14. Add #
15. Get # (all tweets with a specific tag)
16. Like
17. Likes Count
18. Likes List
19. Get Trending Tweets
20. Send Message
21. Get Messages From User
22. Get Message Senders List

### EER Diagram

## CORE TABLES

### USER

|  |
| --- |
| CREATE TABLE twitter.user(  FIRSTNAME NVARCHAR(30) NOT NULL,       LASTNAME NVARCHAR(30) NOT NULL,   ID NVARCHAR(20) NOT NULL,       PASSWORD\_HASH NVARCHAR(128) NOT NULL,       BORN DATE NOT NULL,       JOINED       DATE NOT NULL,       BIOGRAPHY NVARCHAR(64) NOT NULL,       PRIMARY KEY (ID)) |
|  |

### TWEET

|  |
| --- |
| CREATE TABLE twitter.tweet(     ID INT AUTO\_INCREMENT NOT NULL,     USER\_ID NVARCHAR(20)       NOT NULL,     CONTENT NVARCHAR(256)      NOT NULL,     POSTAGE DATE               NOT NULL,     PRIMARY KEY (ID),     FOREIGN KEY (USER\_ID) REFERENCES twitter.user(ID)) |

### MESSAGE

|  |
| --- |
| CREATE TABLE twitter.message(     SENDER\_ID NVARCHAR(20) NOT NULL,     RECEIVER\_ID NVARCHAR(20) NOT NULL,     TWEET\_ID INT,     CONTENT NVARCHAR(256),     POSTAGE DATE,     TYPE INT NOT NULL,     FOREIGN KEY (SENDER\_ID) REFERENCES twitter.user(ID),     FOREIGN KEY (RECEIVER\_ID) REFERENCES twitter.user(ID),     FOREIGN KEY (TWEET\_ID) REFERENCES twitter.tweet(ID))      /\*  TYPE == 0        TWEET  TYPE == 1 CONTENT \*/ |

### HASHTAG

|  |
| --- |
| CREATE TABLE twitter.hashtag(     TWEET\_ID INT NOT NULL,     TAG NVARCHAR(6) NOT NULL,     FOREIGN KEY (TWEET\_ID) REFERENCES twitter.tweet(ID)) |

## MAPPING

### LOGIN

|  |
| --- |
| CREATE TABLE twitter.login(     ID NVARCHAR(20) NOT NULL,     TIME DATE NOT NULL,     FOREIGN KEY (ID) REFERENCES twitter.user(ID)) |

### FOLLOWER

|  |
| --- |
| CREATE TABLE twitter.follower(  SRC\_ID NVARCHAR(20),     DEST\_ID NVARCHAR(20),     PRIMARY KEY (SRC\_ID, DEST\_ID),     FOREIGN KEY (SRC\_ID) REFERENCES twitter.user(ID),     FOREIGN KEY (DEST\_ID) REFERENCES twitter.user(ID)) |

### BLOCKED

|  |
| --- |
| CREATE TABLE twitter.blocked(     SRC\_ID NVARCHAR(20),     DEST\_ID NVARCHAR(20),     PRIMARY KEY (SRC\_ID, DEST\_ID),     FOREIGN KEY (SRC\_ID) REFERENCES twitter.user(ID),     FOREIGN KEY (DEST\_ID) REFERENCES twitter.user(ID)) |

LIKED

|  |
| --- |
| CREATE TABLE liked(  TWEET\_ID INT NOT NULL,     USER\_ID NVARCHAR(20) NOT NULL,     PRIMARY KEY (TWEET\_ID, USER\_ID),     FOREIGN KEY (TWEET\_ID) REFERENCES twitter.tweet(ID),     FOREIGN KEY (USER\_ID) REFERENCES twitter.user(ID)) |

### REPLY

|  |
| --- |
| CREATE TABLE twitter.reply(     ID INT AUTO\_INCREMENT NOT NULL,     TWEET\_ID INT NOT NULL,     REPLY\_ID INT NOT NULL,     PRIMARY KEY (ID),     FOREIGN KEY (TWEET\_ID) REFERENCES twitter.tweet(ID),     FOREIGN KEY (REPLY\_ID) REFERENCES twitter.tweet(ID)) |

## FUNCTIONS

### Register

|  |
| --- |
| INSERT INTO twitter.user VALUES('Bardia', 'Ardakanian', 'BardiArdakanian', SHA2('11111111', 256), '2001-09-02', '2021-02-05', 'CE AUT');  INSERT INTO twitter.user VALUES('Sina', 'Amini', 'Trueshinnyjesus', SHA2('22222222', 256), '2000-05-12', '2021-02-05', 'My name is jeff');  INSERT INTO twitter.user VALUES('Mohammad', 'Salehi', 'M\_salehi\_V', SHA2('33333333', 256), '1999-11-23', '2021-02-05', 'Mohammad Salehi son of Reza');  INSERT INTO twitter.user VALUES('Keivan', 'Ipchi', 'keivanIphci', SHA2('444444444', 256), '2002-04-15', '2021-02-05', 'Relationship status: Neftlix and icecream');  INSERT INTO twitter.user VALUES('Amirhossein', 'Najafi', 'AmirNajafi21', SHA2('55555555', 256), '1998-01-03', '2021-02-05', 'It was too long go read it yourself:)');  INSERT INTO twitter.user VALUES('Omid', 'Dokhanchi', 'Stalin00', SHA2('66666666', 256), '2003-03-12', '2021-02-05', 'hail hentai'); |

### Login

|  |
| --- |
| SELECT \*  FROM twitter.user  WHERE ID='bardiArdakanian' AND PASSWORD\_HASH=SHA2('11111111', 256); |

### Login History

#### Adding new record

|  |
| --- |
| INSERT INTO twitter.login VALUES('bardiArdakanian', current\_date()) |

#### Check history

|  |
| --- |
| SELECT \* FROM twitter.login ORDER BY TIME DESC |
| Post  |  | | --- | | INSERT INTO twitter.tweet (USER\_ID, CONTENT, POSTAGE)  VALUES('bardiArdakanian', 'I just took an exam... Im gonna be unemployed:)', current\_date()) | |
|  |

### Get Personal Tweets

|  |
| --- |
| SELECT \* FROM twitter.tweet WHERE USER\_ID='bardiArdakanian' |

### Follow

|  |
| --- |
| INSERT INTO twitter.follower VALUES('bardiArdakanian', 'Stalin00') |

### Unfollow

|  |
| --- |
| DELETE FROM twitter.follower  WHERE SRC\_ID='bardiArdakanian' and DEST\_ID='Stalin00' |

### Block

|  |
| --- |
| INSERT INTO twitter.blocked VALUES('keivanIphci', 'M\_salehi\_V') |

### Unblock

|  |
| --- |
| DELETE FROM twitter.blocked  WHERE SRC\_ID='bardiArdakanian' and DEST\_ID='Stalin00' |

### Timeline

|  |
| --- |
| SELECT tweet.USER\_ID, tweet.CONTENT, tweet.POSTAGE  FROM twitter.tweet, twitter.follower, twitter.blocked WHERE (follower.SRC\_ID = blocked.SRC\_ID        and follower.DEST\_ID != blocked.DEST\_ID)   and (follower.DEST\_ID = tweet.USER\_ID)        and (follower.SRC\_ID = 'bardiArdakanian'); |

### User activity

|  |
| --- |
| SELECT T.USER\_ID, T.CONTENT, T.POSTAGE FROM twitter.tweet AS T WHERE (T.USER\_ID = 'M\_salehi\_V')  and not exists (  SELECT \*   FROM twitter.blocked as B  WHERE B.SRC\_ID = 'M\_salehi\_V' and B.DEST\_ID = 'Stalin00') |

### Add Reply

|  |
| --- |
| INSERT INTO twitter.reply(TWEET\_ID, REPLY\_ID) SELECT 6,11 WHERE NOT EXISTS (  SELECT \*      FROM twitter.blocked AS B, twitter.tweet AS T1, twitter.tweet AS T2     WHERE (T1.ID = 10 AND T1.USER\_ID = B.SRC\_ID)      AND (T2.ID = 11 AND T2.USER\_ID = B.DEST\_ID) ) |

### Show Replies

|  |
| --- |
| SELECT \*  FROM twitter.reply AS R, twitter.tweet AS T1, twitter.tweet AS T2 WHERE (R.TWEET\_ID = T1.ID) AND (R.REPLY\_ID = T2.ID) AND TWEET\_ID=6 AND NOT EXISTS (  SELECT \* FROM twitter.blocked AS B  WHERE (T1.USER\_ID = B.SRC\_ID AND T2.USER\_ID = B.DEST\_ID)        OR (T1.USER\_ID = B.DEST\_ID AND T2.USER\_ID = B.SRC\_ID) ) |

### Add #

|  |
| --- |
| INSERT INTO twitter.hashtag VALUES(11, '#fffff') |

### Get #

|  |
| --- |
| SELECT USER\_ID, CONTENT, POSTAGE  FROM twitter.hashtag AS TAG, twitter.tweet AS T WHERE TAG.TWEET\_ID = T.ID AND TAG.TAG = '#abcde'  AND T.USER\_ID = 'bardiArdakanian' AND NOT EXISTS( SELECT \*  FROM twitter.blocked AS B WHERE B.SRC\_ID = T.USER\_ID AND B.DEST\_ID = 'M\_salehi\_V') |

### Like

|  |
| --- |
| INSERT INTO twitter.liked SELECT 2,'M\_salehi\_V' WHERE NOT EXISTS( SELECT \* FROM twitter.blocked AS B, twitter.tweet AS T WHERE T.ID = 2 AND T.USER\_ID = B.SRC\_ID AND B.DEST\_ID = 'M\_salehi\_V') |

### Likes Count

|  |
| --- |
| SELECT COUNT(USER\_ID)  FROM twitter.liked AS L WHERE L.TWEET\_ID = 6  AND NOT EXISTS( SELECT \*  FROM twitter.blocked AS B, twitter.tweet AS T WHERE T.ID = 6 AND T.USER\_ID = B.SRC\_ID AND B.DEST\_ID = 'Stalin00') |

### Likes List

|  |
| --- |
| SELECT \* FROM twitter.liked AS L WHERE L.TWEET\_ID = 6  AND L.USER\_ID NOT IN ( -- remove blocked account SELECT B.SRC\_ID FROM twitter.blocked AS B JOIN twitter.tweet AS T ON B.DEST\_ID = T.USER\_ID WHERE L.TWEET\_ID = T.ID) AND NOT EXISTS( -- display nothing if the account has been blocked SELECT \* FROM twitter.blocked AS B, twitter.tweet AS T WHERE T.ID = 6 AND T.USER\_ID = B.SRC\_ID AND B.DEST\_ID = 'bardiArdakanian') |

### Get Trending Tweets

|  |
| --- |
| SELECT COUNT(TWEET\_ID), T.ID, T.USER\_ID, T.CONTENT, T.POSTAGE FROM twitter.liked AS L JOIN twitter.tweet AS T ON T.ID = L.TWEET\_ID GROUP BY L.TWEET\_ID ORDER BY COUNT(TWEET\_ID) DESC |

### Send Message

|  |
| --- |
| INSERT  INTO twitter.message (SENDER\_ID, RECEIVER\_ID, TWEET\_ID, POSTAGE, TYPE) SELECT 'bardiArdakanian','M\_salehi\_V', 13, current\_date(), 0 FROM twitter.tweet AS T WHERE T.ID = 13 AND 'M\_salehi\_V' NOT IN (  SELECT B.DEST\_ID   FROM twitter.blocked AS B  WHERE T.USER\_ID = B.SRC\_ID) AND NOT EXISTS(  SELECT \* FROM twitter.blocked AS B  WHERE (B.SRC\_ID = 'bardiArdakanian'        AND B.DEST\_ID = 'M\_salehi\_V') OR (B.SRC\_ID = 'M\_salehi\_V'        AND B.DEST\_ID = 'bardiArdakanian')) |

#### Delete Message

|  |
| --- |
| DELETE FROM twitter.message WHERE SENDER\_ID = 'Trueshinnyjesus' AND RECEIVER\_ID = 'M\_salehi\_V' |

### Get Messages From User

|  |
| --- |
| SELECT \* FROM twitter.message AS M, twitter.tweet AS T WHERE M.RECEIVER\_ID = 'M\_salehi\_V' AND T.ID = M.TWEET\_ID AND 'M\_salehi\_V' NOT IN (  SELECT B.DEST\_ID   FROM twitter.blocked AS B  WHERE T.USER\_ID = B.SRC\_ID) ORDER BY M.POSTAGE DESC |

### Get Message Senders List

|  |
| --- |
| SELECT SENDER\_ID, COUNT(SENDER\_ID) AS NEW FROM twitter.message AS M WHERE M.RECEIVER\_ID = 'M\_salehi\_V' GROUP BY SENDER\_ID ORDER BY POSTAGE DESC |

## EER Diagram

